分散型システムにおける相互排除のためのトークンを用いたスキームについて(計算機構に関する数学的基礎理論とその応用)

AUTHOR(S):

室, 章治郎; 加藤, 直樹; 箕浦, 敏美; 長谷川, 利治

CITATION:

室, 章治郎 ...[et al]. 分散型システムにおける相互排除のためのトークンを用いたスキームについて(計算機構に関する数学的基礎理論とその応用). 数理解析研究所講究録 1983, 494: 114-125

ISSUE DATE:

1983-06

URL:

http://hdl.handle.net/2433/103572

数理解析研究所講究録
494巻 1983年 114-125

分散型システムにおける相互排除のための トークンを用いたスキームについて

京都大学・工　室　章治郎(Shojiro Muro)
神戸商科大学　加藤　直樹(Naoki Katoh)
オレゴン州立大学　箕浦　敏美(Toshimi Minoura)
京都大学・工　長谷川利治(Toshiharu Hasegawa)

1. まえがき

相互排除(mutual exclusion)問題は，もともと，集中型計算機システムにおいて，干渉しあう並行プロセスが，資源（変数，ファイル等）を共用するために必要な排他制御を実現する時間的な同期化規則として規定された[DIJK-65]．並行プロセスは，臨界領域(critical region，以後，CRと略す)とよばれる構造化された(プログラム)文の中でのみ共用資源を参照し書きかえることができ[DIJK-65, HANS-73]，CR文を実行できる（実行がおわる）ことをCR内に入いる（CRから出る）という。同じ共用資源を参照するCRが時間的な相互排除を実現し，プロセス間の実行が滞りなく進むために，並行プロセスは各共用資源に関する次の臨界領域条件を満たさなけ

ればならない。

(i) CRに入いろうとしたプロセスは，その要求が有限時間内に受け付けられる。

(ii) プロセスは，有限時間内でCRから出る。

(iii) 一時には高々一つのプロセスしかCRに入いれない（相互排除の要請）。

分散型計算機システムは，1970年代におけるコンピュータの低価格化，通信技術の高度な発達という情況のもとで出現し，その制御法については現在も活発に研究が進められている。分散型システムにおいては，ある共有資源び（特に，そのコピーが複数個のサイトに置かれている場合）の臨界領域条件を満たすために，どのサイトからも参照できる共有メモリー（または，バッファー）を置き，（コピーも含めた）資源びに関する集中制御を行なうことは，システムの分散制御の思想からも好ましくなく，サイト間のメッセージ交換のみによる制御法が一般的に用いられている。今までに，論理時計を用いたLamportの方法[LAMP-78]をはじめとして，分散型システムに対して多くの相互排除アルゴリズムの提案およびその解析が行なわれているが[LELA-77, REED-79, PETE-79, RABI-80, RICA-81, SUZU-82]，特に，[SUZU-82]では，その論文で提案されたアルゴリズムも含めいくつかのアルゴリ

ズムについて，メッセージ交換により生ずるパスの構造，メッセージ数およびその種類を比較しながら議論してあり興味深い。

ところで，今までに提案された分散型システムに対する相互排除アルゴリズムでは，システム内に故障が発生した場合のアルゴリズムの継続的な実行に関してはあまり論じられなかった。本稿では，最初に，トークンの譲渡を基本とし，メッセージ交換によって実現される分散型システムの相互排除アルゴリズムを提案する。次に，トークンがシステムの故障により失なわれる可能性のある場合に，トークンを正常に再生するアルゴリズムについて述べる。さらに，システム内のサイトまたは通信路に故障が発生した場合における，提案したアルゴリズムの継続的実行について概略のみ述べる。

2. システムモデル

本稿で議論の対象とする分散型システムは，故障の生じていない場合，次の条件を満たすものとする。(1)システムはN個のサイトからなり，サイト名を1, 2, …, Nとする。(2)各サイトは共通にアクセスできる記憶部分をもたないものとする。(3)任意の二つのサイト間には，通信経路（二つのサイトi, jを直接結ぶ通信回線を通信路(i, j)といい，いくつかの通信

路をつないだ通信回線を通信経路とよぶことにする）が存在し，各サイトはその通信経路を通したメッセージ交換により，他のサイトと通信することができる．(4)発送されたメッセージが発送順に目的のサイトに着くという仮定はない．(5)2サイト間の通信遅延は有限であると仮定するが，遅延の上限は設定しない．(6)各サイトには，繰返しCRに入る一つのプロセスがあり，各プロセスはCRへ入る要求を一旦出したら，そのCR要求が認可され，CRへ入いり，CRから出るまで次のCR要求を出すことができない．但し，各プロセスがCR内に入っている時間は有限であるとする．

以上の条件のもとで，次節で制御トークン(control token)をサイト間で次々と譲渡(transfer)して，相互排除を実現するアルゴリズムCTTA(control token transfer algorithm)を記述する．

ここで，サイト間のトークンの譲渡に関して次の仮定をする．(7)トークンがあるサイトを出て，任意の他のサイトに行き着くまでの時間は有限である．

3.アルゴリズムCTTA

アルゴリズムCTTAでは，各共有資源Vに関してサイト間にトークンが1個存在し，トークンを有するサイトのプロセス

のみが資源rに関してCRに入ることができる。サイトiにn回目のCR要求が発生したとき，他の各サイトjに送られるメッセージはCRREQ(i,j,n)の形をしており，nをサイトiの要求番号とよぶ。各サイトiには，長さNの配列LR_iがあり，各j≠iに対して$LR_i(j)$はサイトiがその時点までに受け取ったサイトjからのCRREQメッセージの最大の要求番号が格納され，$LR_i(j)<n$なるCRREQ(j,i,n)がiサイトiに届くたびに$LR_i(j)\leftarrow n$と更新される。$LR_i(i)$は，サイトiが最後に送ったCRREQの要求番号を記憶する。トークンは長さNの配列HRをもち，HR(i)はその時点までにサイトiがCRに入った回数を示す。サイトiはHR(i)の値を保持しているものとする(次節のトークン再生アルゴリズムで必要)。サイトiにCR要求が発生すると，$LR_i(i)\leftarrow LR_i(i)+1$と更新し，メッセージCRREQ(i,j,$LR_i(i)$)を他の各サイトjに送る。

CR要求を出していたサイトiにトークンが訪れた場合，サイトiはCRに入いる。CRから出るとき，$HR(i)\leftarrow HR(i)+1$と更新し(この時点で，$HR(i)=LR_i(i)$が成立する)，

$$LR_i(j) > HR(j) \qquad (3.1)$$

の条件を満たすjのうちNの剰余のもとでiの次に大きいサイト番号をもつサイトにトークンを譲渡する。

以上のようにしてトークンの譲渡が行なわれて，CR要求を

出していたサイトが次々とCRに入っていく.

(注意3.1) アルゴリズムCTTAは, 次にCRに入いるサイトの決定の仕方が異なることを除き, [SUZU-82]のAlg 1と本質的にはほぼ同じであり, [SUZU-82]のメッセージ"YOURTURN"をトークンと対応させることができる. [SUZU-82]では, 条件式(3.1)を満たすサイトを調べるたびに, 条件を満たすサイトをQとよばれるqueueに記憶し, Qに入ったサイト順にトークンが譲渡される. システムの満たす条件(4)を考えると, このようなQを用いたとしても, CR要求を出した時間的順序とCRに入いるサイトの順序とは必ずしも一致せず, また, トークンが失なわれQの情報が失なわれた場合の損失の大きさを考え, アルゴリズムCTTAでは, トークンの譲渡に関して簡単な方法を採用した。 □

(注意3.2) 条件式(3.1)を満たすiが存在しないとき, トークンはサイトiに停まっているものとする。サイトiも含めて, どれかのサイトがCR要求のメッセージを出せば, 2節の条件(5)より, 有限時間でサイトiにその要求が着き, トークンの譲渡が行なわれる。 □

共有資源Rに関して, アルゴリズムCTTAではトークンを有するサイトのみがCRに入いれること, さらに, システムの満たす条件(5), (6), (7) および注意3.2より明らかに次の定理

が成立する.

定理3.1. アルゴリズムCTTAは,各共有資源に関して,臨界領域条件を満たす。 □

アルゴリズムCTTAでサイトiがCR要求を出してCRに入るまでに必要なメッセージ数は,[SUZU-82]のAlg 1 同様Nであり,[RICA-81]のアルゴリズムのメッセージ数の半数である。また,メッセージの種類は,$LR_j(i)$, $HR(i)$のiが可算無限個の値を取ることより,可算無限種必要である。

4.トークン再生アルゴリズム

2節で述べたシステムの満たすべき条件下では,トークンがシステム内で失なわれること(以後,トークンロスという)は考えられない。ところが各サイトや通信路に故障が生じるとトークンロスの可能性があり,トークンの再生が必要となる。各サイトが,例えばCR要求を出して一定時間待ってもトークンが譲渡されて来ないというような情況のもとで,トークンロスの疑いをもった場合,トークンの存在を調べ,もし,トークンが失なわれていたらトークンを1個再生するアルゴリズムREGENERATEを実行することができる。

各サイトiは,サイトiの年齢とよばれる非負整数CG(i)を,また,トークンは,トークン年齢とよばれる非負整数TG

を保持し，これらの値は0に初期値設定されているものとする。トークン（再生した場合には，再生後のトークン）を保持しているサイトiに対して，常に$TG=CG(i)$が成立するようにアルゴリズムが働く。

あるサイトiがトークンロスの疑いをもった場合，新しいトークン年齢の候補g_iを次式で定める。

$g_i=\min\{l \mid l=pN+i$（pは非負整数）and $l>CG(i)\}$。

$CG(i)$を$CG(i)\leftarrow g_i$と更新し，他の各サイト$j(\neq i)$に$CG(j)$を更新するためのメッセージ$UDCG(i,j,g_i)$を送り，それらのサイトから返事が返ってくるのを待つ。もし，全てのサイトjから$ACK(j,i)$が返ってきたらサイトiでトークンを再生し，TGを$TG\leftarrow g_i$と更新する。もし，あるサイトjから$NACK(j,i)$が返ってきたら，サイトiでのトークンの再生，TGの更新は行なわない。

メッセージ$UDCG(i,j,g_i)$を受け取った各サイトjでは，もし，$g_i<CG(j)$ならばサイトiに$NACK(j,i)$を返す。$g_i>CG(j)$の場合，次の二通りに分かれる。もし，トークンがサイトjにあるならば，$CG(j)\leftarrow g_i$，$TG\leftarrow g_i$と更新して，$NACK(j,i)$をサイトiに返す。トークンがサイトjになければ，$CG(j)\leftarrow g_i$と更新し，$ACK(j,i)$を$HR(j)$を付して（サイトiでトークンを再生するために必要）サイトiに返す。

以上のアルゴリズムREGENERATEにおいて，トークンが複数個できるのを防ぐために，アルゴリズムCTTAに次の修正を加える必要がある。

サイトiにトークンが譲渡され，そのトークン年齢がTGであるとする。もし，TG<CG(i)なら譲渡されてきたトークンを除去する。それ以外の場合，サイトiはCRへ入いり，以後アルゴリズムCTTAに従ってトークンを譲渡していく。

各サイトで定義されるトークン年齢の候補g_iの値が各サイトごとに異なり，しかも単調増加すること。サイトiがトークンを生成できるためには，g_iが全てのCG(j)(j≠i)より大きくなければならないこと。さらに，トークンが通信路上にあるときにどこかのサイトで新たなトークンが生成された場合，通信路のトークンがその後最初に訪れたサイトで，アルゴリズムCTTAの修正により，除去されることより次の定理を示すことができる。

定理4.1. アルゴリズムREGENERATEの実行後必ずトークンが存在し，しかも2個以上トークンが作られることはない。

□

5. 故障が生じた場合のアルゴリズムCTTAの実行

最初に，本稿で対象とする故障の種類と故障時にシステムが満たすと仮定してよい条件について[GARC-82]を参考に述べる。

故障の種類：システム内の故障の種類として，サイトの故障と通信路の故障を考える。故障していないサイトを活性サイトといい，通信路の故障が原因であるZサイト間を結ぶ全ての通信経路が正常なメッセージの伝送を行なえない場合，サイトの分割が起ったという．

故障時にシステムが満たす条件：サイトiまたは通信路(i,j)の故障が発生した場合，サイトiまたは通信路(i,j)の両端i,j（通信路(i,j)の故障中，サイトi,jは故障しないものとする）がそれぞれ責任をもち，有限時間内で他のサイトに故障の発生が知らされる（実現法は[MINO-82]に詳しい）。同時に故障中であるサイトの最大数は小さく，サイトの分割は高々二分割までである。サイトiに故障が起った場合，サイトiはメッセージを受信し，他のサイトに伝送することはできるが，メッセージの内容に応じて，HR(i), CG(i)および配列LRiの値は一切更新できない。また，これらの値は故障時に失なわれない（失なわれても，システム内のログ[GARC-82]を用いて再生可能である）。サイトおよび通信路の故障は有限時間で回復する。

サイトまたは通信路が故障した場合，故障発生時にトークンが失なわれなければ，活性サイト間では臨界領域条件が満たされるために，それぞれの故障の場合に対してアルゴリズ

ムCTTA またはREGENERATEを以下のように修正しなければならない。

サイトの故障時のアルゴリズムの修正：サイトiが故障前にCR要求を出したために，トークンがサイトiに故障中に譲渡されてきたら，そのトークンをサイトiと通信路で結ばれている任意のサイトjに譲渡する。また，サイトiの故障中，$LR_i(j)$ $(j=1,2,\cdots,N)$ の値が更新されないため，各$LR_i(j)$の値が現在の$LR_j(j)$の値に比べて非常に小さく，サイトiの故障回復直後にサイトiにトークンがある場合，式(3.1)を満たすサイトが一つもなくトークンの動きが止まる場合がある。そのため，CR要求を出している各サイトjは，再度同じ$LR_j(j)$の値をもってCRREQ(i,j,$LR_j(j)$)を出してもよいものとする。また，サイトの故障中，他の任意のサイトjからのメッセージUDCGに対しては，無条件にNACKを返す。

通信路(i,j)の故障によるサイト分割時のアルゴリズムの修正：サイトi(j)は，サイトi(j)と通信経路で結ばれているサイトの集合に，サイトの分割が起った旨のメッセージPARTITION(i)(PARTITION(j))を送る。そのメッセージを受け取った各サイトは，アルゴリズムCTTAは続行するが，アルゴリズムREGENERATEは実行せず，また，他の任意のサイトからのメッセージUDCGを受け取っても無条件に

NACKを返す。サイトの分割により，CRREQが全てのサイトに届かない場合があり，サイトの故障の時と同様再要求を許す。

6. むすび

本稿では，システムの故障によりトークンが失なわれることも考慮した臨界領域条件を満たすアルゴリズムについて論じた。現在，故障時にシステムが満たす条件等を検討中である。

[謝辞] 末筆ながら，室，加藤が日頃御指導頂き，本稿に対してもコメントを頂いた京都大学矢木俊秀助教授に感謝する。

参考文献

[DIJK-65] Dijkstra, E.W., "Solution of a problem in concurrent programming control", CACM, 8, 9 (1965), 569.

[GARC-82] Garcia-Molina, H., "Reliability issues for fully replicated distributed databases", IEEE Computer (Sept. 1982), 34-42.

[HANS-73] Hansen, P.B., "Operating system principles", Prentice - Hall, Inc. (1973).

[LAMP-78] Lamport, L., "Time, clock, and the ordering of events in a distributed system", CACM, 21, 7 (1978), 558-565.

[LELA-77] LeLann, G., "Distributed system - Toward a formal approach", IFIP 77 (1977), 155-160.

[MINO-82] Minoura, T., "Resilient extended true-copy token scheme for a distributed database system", IEEE Tr. on SE (1982), 173-189.

[PETE-79] Peterson, G.L., "Concurrency and complexity", TR-59, Department of Computer Science, The University of Rochester (1979).

[RABI-80] Rabin, M.O., "N-process synchronization by $4 \cdot \log_2 N$-valued shared variable", Proc. 21st IEEE Anual Symp. on FOCS (1980), 407-410.

[RICA-81] Ricart, G. and Agrawara, A.K., "An optimal algorithm for mutual exclution in computer networks", CACM, 24, 1 (1981), 9-17.

[REED-79] Reed, D.P. and Kanodia, R.K., "Synchronization with eventcounts and sequences", CACM, 22, 2 (1979), 115-123.

[SUZU-82] Suzuki, I and Kasami, T., "An optimality theory for mutual exclusion in computer networks", Proc. 3rd Inter'l Conf. on DCS (1982), 365-370